hp OpenView
Service Quality Manager

Getting Started Guide

Edition: 1.2

for the HP-UX and Microsoft Windows Operating Systems

March 2005

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company

United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

©Copyright 2000-2004 Hewlett-Packard Company, all rights reserved.

No part of this document may be copied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

Netscape is a U.S. trademark of Netscape Communications Corporation.

NMOS™ is a trademark of RiverSoft Technologies Limited.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

Oracle7™ and Oracle7 Server™ are trademarks of Oracle Corporation, Redwood City, California.

PostScript® is a trademark of Adobe Systems Incorporated.

Riversoft™ is a trademark of RiverSoft Technologies Limited.

UNIX® is a registered trademark of The Open Group.

Windows® and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Contents

Preface

This document describes how to implement service quality management with OpenView Service Quality Manager (SQM). It gives an overall view of what OpenView SQM operators do at the different stages of the Service Level Agreement (SLA) lifecycle.

Intended Audience

This document addresses the following audience:

- Solution architects and service designers

- Service operators

Prerequisite Reading

This document assumes that you have read the *OpenView Service Quality Manager Overview*.

Supported Software

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

The supported software referred to in this document is as follows:

Product Version	Operating System
OpenView Service Quality Manager 1.2	HP-UX 11.11 Windows XP

Typographical Conventions

`Courier` Font:

- Source code and examples of file contents.

- Commands that you enter on the screen.

- Path names

- Keyboard key names

Italic Text:

- Filenames, programs and parameters.

- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

Associated Documents

The OpenView SQM documentation set includes the following documents:

- *OpenView SQM SLA Monitoring UI User's Guide*
- *OpenView SQM Service Designer UI User's Guide*
- *OpenView SQM SLA Administration UI User's Guide*
- *OpenView SQM Overview*
- *OpenView SQM Getting Started Guide*
- *OpenView SQM Information Modeling Reference Guide*
- *OpenView SQM Installation Guide*
- *OpenView SQM Administration Guide*
- *OpenView SQM Reference Guide for Oracle Use*
- *OpenView SQM Datamart User's Guide*
- *OpenView SQM Reporting Customization and User's Guide*

Refer to the following document for useful reference information:

- *TeleManagement Forum Service Level Agreement Management Handbook, v 1.5.*

Support

Please visit our HP OpenView web site at: HP OpenView

There you will find contact information as well as details about OpenView products, services and support.

The OpenView support area of the OpenView web site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

Chapter 1

Introduction to Service Quality Management

This chapter introduces the SLA lifecycle. The SLA lifecycle describes the main functional components needed to manage service level and SLAs. It includes the following phases:

- **Service design phase**

  In the service design phase, you identify all the technical resources needed to provide an end-to-end service. You model each of these resources as a service component and describe the service components with a set of parameters. The service model allows you to use these parameters to evaluate the service quality you deliver.

  The result of this phase is a unique service class, broken down into service components that provide the relevant service quality parameters.

  The service design phase is a critical, stand-alone process that aims to create a common definition of services, service parameters, and service level objectives. These common definitions can then be shared between the service provider, users, suppliers, and partners.

- **Class of service definition phase**

  Once you have described a service and selected the service parameters used to evaluate the quality of service, you must define a standard class of service, or **service level**, for that service.

  A service level consists of a set of objectives defined for the service parameters. Service parameters are evaluated directly from measurements made in the service infrastructure, or computed from the evaluated parameters. Different service levels correspond to different objectives set for the same service parameters.

  Service levels do not need to be defined for a specific customer, but instead can be a part of a general service definition.

- **Service instantiation phase**

  Before the service instantiation phase, a negotiation and sales phase occurs that leads to an SLA contract signature for a particular service. Negotiation and sales is outside the scope of the SLA lifecycle.

  In the service instantiation phase, a new service needs to have its quality monitored. For example, a service model needs to be instantiated when a customer buys a new service. Service quality management begins as soon as you launch the service.

  This phase results in a service instance and triggers the collection of service quality parameters.

- **SLA creation phase**

  In this phase, the service provider creates an SLA for a specific service instance or a group of service instances, depending on the contract. The service provider also identifies the threshold values of the service level for each service quality parameter managed for the SLA.

  This phase results in an SLA that:

  Associates the service instance(s) with the service level and a customer (for customer SLAs)

  Associates the service instance(s) with the service level and an internal department (for operational SLAs)

- **Service monitoring phase**

  In this phase, the service provider monitors the service and the service level agreements in real-time.

  The goal of the service monitoring phase is to avoid SLA violations by rapidly responding to service degradations.

- **Service reporting phase**

  In the service level reporting phase, the service provider generates reports about service quality.

  This phase assesses service quality on a long-term basis (monthly or quarterly) to analyze service trends or to provide customer service quality reports.

After the generation of reports, the service provider can periodically assess the service quality being provided to a particular service and the overall service quality for all customers. Depending upon new goals and changing customer needs, the service provider can make changes to the service design, completing the SLA lifecycle.

OpenView SQM provides the tools you need to easily manage each phase of the SLA lifecycle.

Chapter 2

Designing a Service Model

This chapter describes the first phase of the SLA lifecycle, service design. During the service design phase, you implement a model for a service quality management solution with OpenView SQM.

This chapter contains the following sections:

- Section 2.1 Before You Begin

- Section 2.2 Modeling Your Service

- Section 2.3 Implementing Your Service Model in XML

2.1 Before You Begin

This section provides the information you need before beginning the service design phase of your service level monitoring solution. This section also describes the service example used throughout this guide to illustrate each step of the SLA lifecycle.

2.1.1 Prerequisites of the Service Design Phase

The design phase assumes that you have:

- Identified the service quality requirements during a planning phase, which is outside the scope of the SLA lifecycle.

- Started to design the service model for service quality management

- Installed the OpenView Service Designer UI

- Installed the OpenView SQM components

2.1.2 Goal of this Phase

The goal of the service design phase is to produce a full service model that meets your needs, and to load this model in OpenView SQM. Once you have provided OpenView SQM with your service model, you create instances during the next phase of the SLA lifecycle, service instantiation.

2.1.3 A Service Design Example

To illustrate the SLA lifecycle, we use the video service introduced in the *OpenView Service Quality Manager Overview*. In this example, an Internet service provider (ISP) wants to create a new video service that allows subscribers to download movies over a wireless LAN and watch them on an iPaq terminal. The ISP wants to offer three levels of service, each priced according to the quality of service provided.

The remainder of this guide builds on this example through each step of the SLA lifecycle.

2.1.4   Overview of the Design Process

The service design phase involves modeling your service, implementing this model, and updating the model in OpenView SQM. You can use the OpenView Service Designer UI to execute each step of the design phase. When you implement your service model, the Service Designer automatically generates XML code. You need to load this XML in OpenView SQM to update the generic service model for your service quality management solution. You provide these files to OpenView SQM using the Command-Line User Interface (CLUI).

If you want to develop your own XML code instead of using the OpenView Service Designer UI, you can design and implement the service model yourself. However, remember that developing code takes time and resources. Your HP representative can provide you with more information about developing a service model directly in XML.

Expressions play an important role in the service model. Expressions describe how to calculate the service parameter values from the service quality data collected. OpenView SQM provides a set of pre-defined expressions for basic calculations. However, if you need an expression not provided by OpenView SQM, you can implement your own expressions in PL/SQL or Java.

Figure 1 describes the steps involved in designing a service model.

Figure 1   **Designing a Service Model**
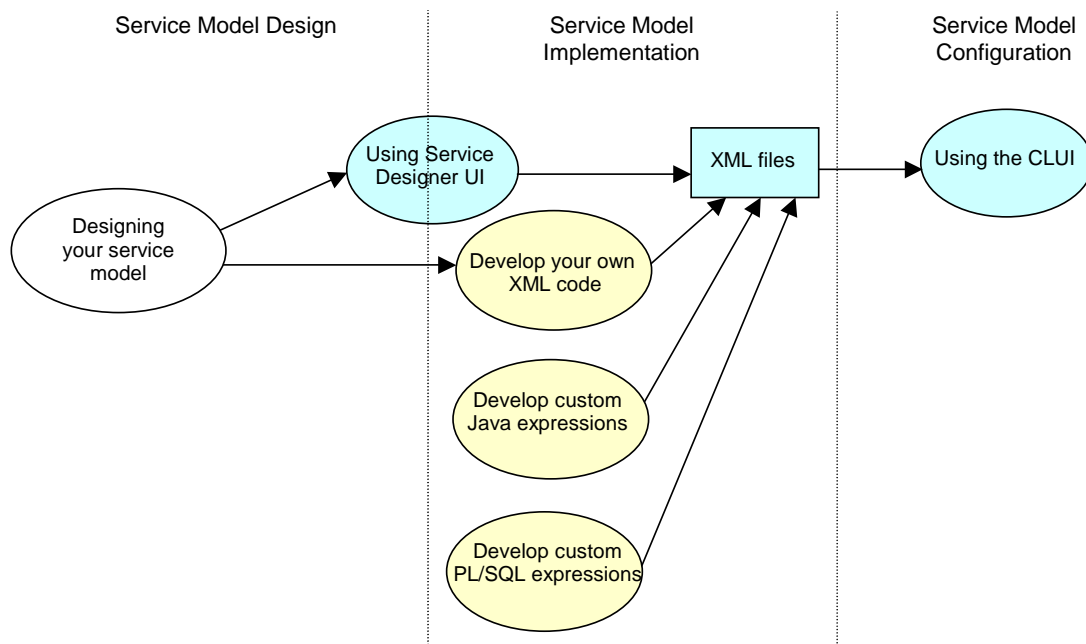
2.2 Modeling Your Service

The decisions you make when you design your service model affect the overall success of your service quality management solution. The service model affects the way you represent, populate, and monitor services.

The service model helps the service provider provide a quality service. This section provides some advice about how to design the service model once you have gathered the general service quality requirements. It describes the tools available to help you in this phase, and the general steps used.

2.2.1 Design from the Top Down: Defining the Service

The following steps are recommended for an efficient modeling of your service:

- Identify the service quality parameters that the SLAs will use.

    For example, the designer of the video service identifies two parameters, NbDwndMovies (the total number of downloaded movies) and OperationalState (the global operational state of the service) as two parameters that the SLAs will be based on for measuring service quality. Table 1 describes these and other parameters identified by the service designer and the data types.

Table 1   Sample Parameters

Parameter Name	Data Type
NbDwndMovies	Integer
OperationalState	Enumeration
Throughput	Float
DownloadDuration	Integer
NbErrors	Integer
Delay	Relative time

- Identify the components of your service.

    Service components generally come from the service infrastructure, the Operation Support System (OSS) systems involved with the service, or applications dedicated to the service. Select service components that bring some benefit to service quality management.

    For example, the designer of the video service decides that the service needs a Web server component to host the video server and the Web server, an Internet Protocol (IP) access component, a Domain Name System (DNS) component, and a video service component (which represents the service itself).

- Associate the service quality parameters with the relevant service components.

    For example, the video service designer associates the NbDwndMovies parameter with the video service component and the OperationalState parameter with each service component.

- Identify the relationships between the service and the service components according to the service quality dependencies.

    For example, the quality of service delivered by the video service depends on the quality of service provided by the Web server component that provider users access to the service.

2.2.2 Design from the Bottom Up: Identifying the Data Feeders and Service Adapters You Need

OpenView SQM manages the set of service quality parameters you define in the service model, as described in the previous section. Data feeders provide the values of the service quality parameters. Identifying the data feeders consists of:

- Identifying the source of service quality values.

  For example, the video service collects information about the number of downloaded movies from a SQL database.

- Characterizing the source of service quality data.

  You must create a thorough description of the data source so that you can identify and implement the service adapters you need. Service adapters are the OpenView SQM components that retrieve all the necessary information from a specific source. The description of the data source includes the technology, the interface, the communication protocol, and the operating system of the source.

- Specifying the parameters retrieved to populate the service parameters in the service model.

  For example, the video service gets the `NbMovies` parameter from the SQL database to populate the `NbDwndMovies` parameter in the service model.

- Defining a Measurement Reference Point naming scheme.

  OpenView SQM retrieves the service quality data from several places, on various service instances. OpenView SQM uses the Measurement Reference Point (MRP) to manage the names of data feeder instances. With the MRP naming scheme, you define the automatic generation of data feeder identifiers using fixed strings combined with custom properties.

2.2.3 Using the OpenView Service Designer UI to Create Your Service Model

The OpenView Service Designer UI uses the Rational Rose™ Modeler Edition product to represent the OpenView SQM object model in Unified Modeling Language (UML). Rational provides a software development platform that specializes in object-oriented analysis, design, and modeling following the UML standard.

The OpenView Service Designer UI helps you design a custom service model that complies with the definitions of the OpenView SQM service model. The UI lets you build a service model graphically so that you can visualize your design.

The OpenView Service Designer UI automatically generates the XML code that you need to implement your service model in OpenView SQM.

The OpenView Service Designer UI is a stand-alone application. Once installed, you can use it immediately. There is no need to install OpenView SQM platform to use OpenView Service Designer UI.

2.2.3.1 Before You Begin

Before designing your service model with the OpenView Service Designer UI, consider the following:

- The service model has to respect the rules imposed by the OpenView SQM object model. For details about the object model, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide*.

- The OpenView Service Designer UI represents the service model as a class diagram.

- Services, service components, and data feeders are pre-defined classes in the OpenView Service Designer UI.

- Parameters and properties are pre-defined attribute stereotypes in the OpenView Service Designer UI.

- The OpenView Service Designer UI represents expressions as sequence diagrams.

2.2.3.2   Creating Your Service Model

The designer of the video service creates a concrete service object model that supports the components it identified earlier. Figure 2 describes the component objects and how they interact.

Figure 2   Video Service Concrete Object Model
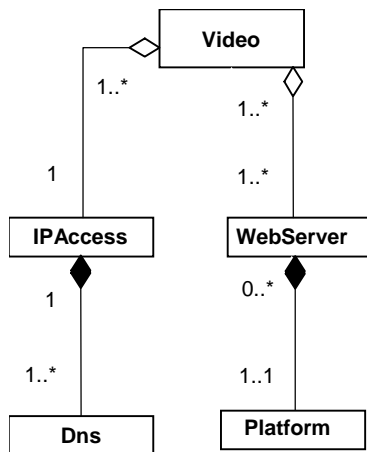
The object diagram describes the relationship between the video service object and its component objects. The numbers in the diagram indicate the **cardinality**, or how many instances of each component object can be present.

The service designer creates a service object and gives the service a description as described in Figure 3.

Figure 3   OpenView Service Designer UI: Defining Video Service Object
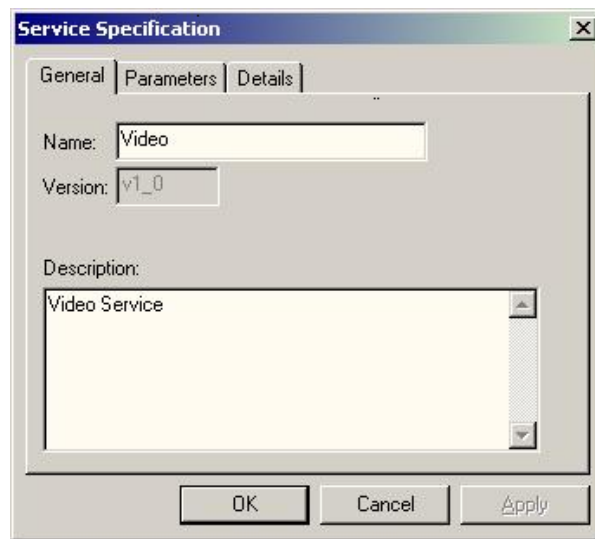
Next, the designer gives the service a set of parameters that OpenView SQM collects, aggregates, and propagates after service deployment. OpenView SQM checks these parameters against the objectives for the customers. If degraded and violated, it triggers alarms.

The designer defines the video service object using the parameters and properties identified in Section 2.2.1. Figure 4 illustrates how parameters and properties are set using the Attributes tab in the Service Designer.

Figure 4   **OpenView Service Designer UI: Specifying Video Service Parameters**
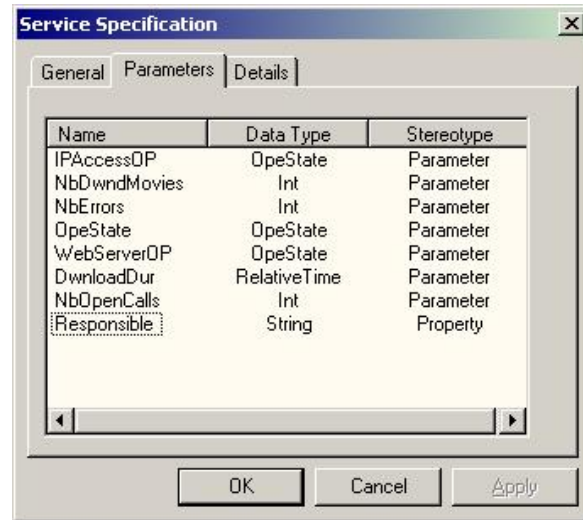
The service designer repeats this process for every component object required by the Video service. When complete, the concrete object model of the Video service appears in the OpenView Service Designer UI as illustrated in Figure 5.

Figure 5   **OpenView Service Designer UI: Video Service Object Model**
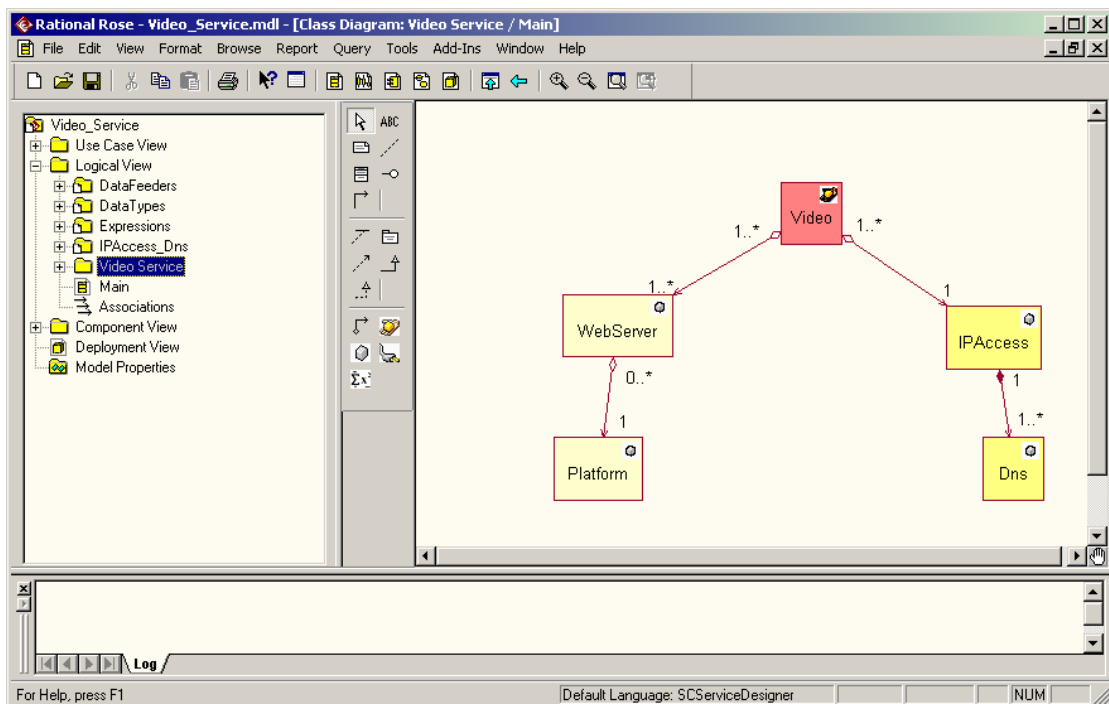
2.2.3.3   Other Features: Expressions and Mapping

You can also use the OpenView Service Designer UI to enhance your service objects.

Usually there is a gap between the service parameters you need and the data feeder parameters retrieved by OpenView SQM. To bridge this gap, you need to define expressions. Expressions define the operations applied to the data feeder parameters to obtain the service parameters values. For more information about expressions, refer to Section 2.2.4.

An ISP can offer services, such as mail and Web services, to other businesses. These businesses, in turn, can give employees and subscribers access to the services. To make this mapping between subscribers and customers, OpenView SQM uses subscriber IDs. The subscriber ID identifies the user of the service in a service quality measure. For more information about mapping subscriber IDs, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide.*

2.2.4 Specifying Expressions

Usually there is a gap between the service parameters you need and the data feeder parameters retrieved by OpenView SQM. To bridge this gap, you need to define expressions. Expressions define the operations applied to the data feeder parameters to obtain the service parameters values.

You can use expressions to calculate service parameters from the service parameters of the same service component or from a directly related service component.

You can also use expressions to specify an election mechanism for deciding between several values of the same parameter so that the real-time calculation uses the selected value. By default, expressions use the last value collected.

OpenView SQM provides some predefined, generic expressions. However, if you need different expressions, you can create your own. For more information about creating custom expressions, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide*.

The following sections describe the two types of predefined expressions included with OpenView SQM.

2.2.4.1 Simple Expressions

Simple expressions make calculations using parameters from one service component or data feeder instance. You can use simple expressions to:

- Calculate a service parameter from one or several parameter(s) of a given data feeder.

  The video service designer uses the `assign_Int` expression to assign the value of the `KbytesRcvd` parameter collected by the DNS data feeder definition to the `ReceivedKbytes` parameter of the DNS component.

- Calculate a service parameter of an upper service component from one or several parameter(s) of a directly related lower service component instance.

  The video service designer uses the `assign_enum` expression to assign the `platform` operational state to the `WebServer` operational state in the service model.

- Calculate a service parameter from one or several parameter(s) within the same service component instance.

  The video service designer uses the `sum_2int` expression to assign the sum of the `ReceivedKBytes` parameter and the `SentKBytes` parameter to the `ExchangedKbytes` parameter.

Figure 6 illustrates this example.

Figure 6   Simple Expressions
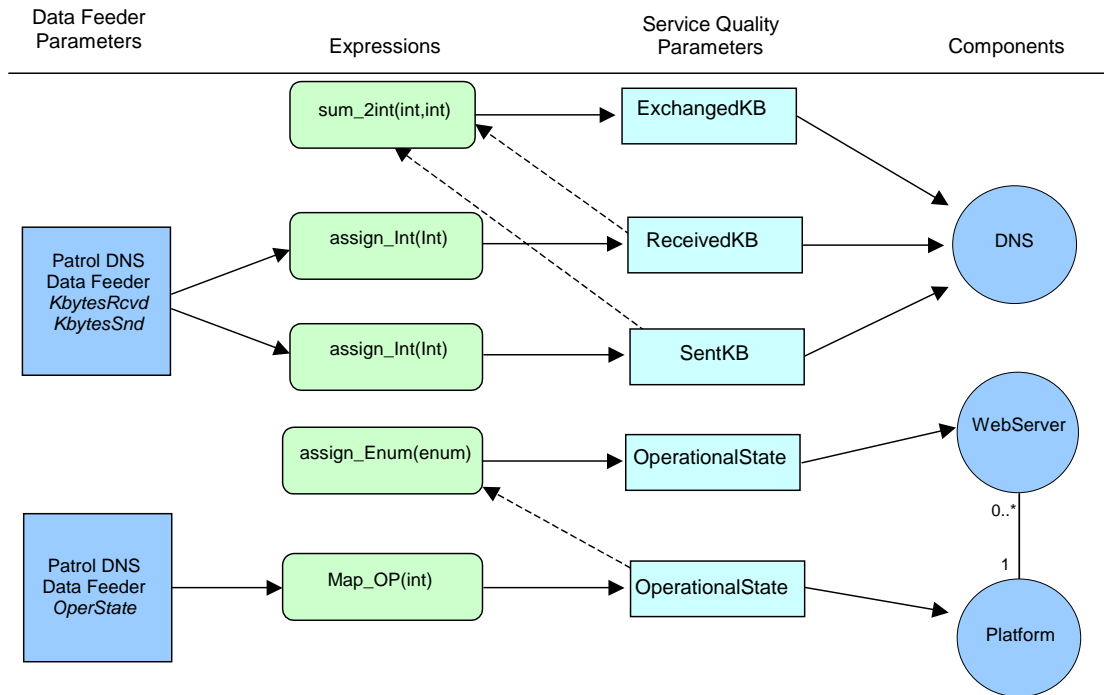
The expressions are selected by the service designer using the OpenView SLA Administration UI as illustrated in Figure 7.

Figure 7   Example Simple Expression
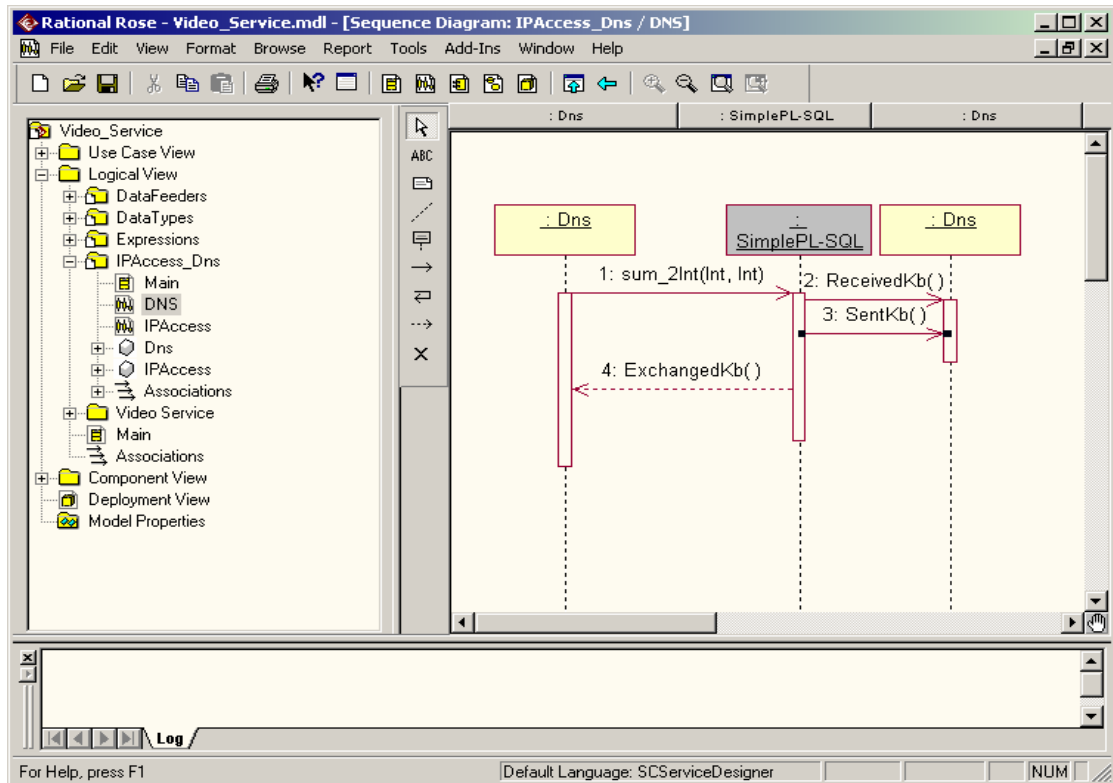
2.2.4.2   Aggregation Expressions

Aggregation expressions make calculations using parameters from more than one service component instance. For example, the ISP wants to calculate the worst operational state of the Web server component. Because you can have more than one instance of the Web server component (cardinality of 1..*), the expression is an aggregation expression. Figure 8 shows the aggregation expression in the definition.

Figure 8   Aggregation Expressions
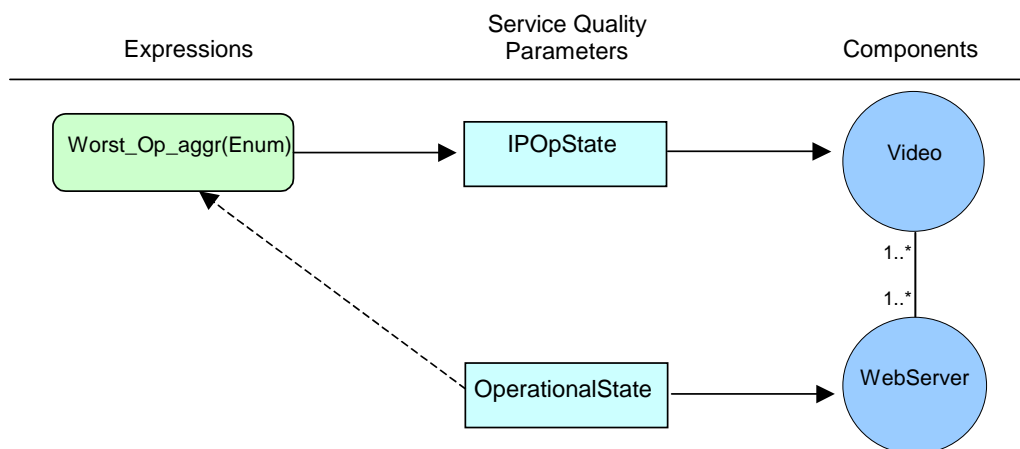
Figure 9 illustrates the aggregation expression when it is used with several instances of the Web server component.

Figure 9   Aggregation Expression with Multiple Instances
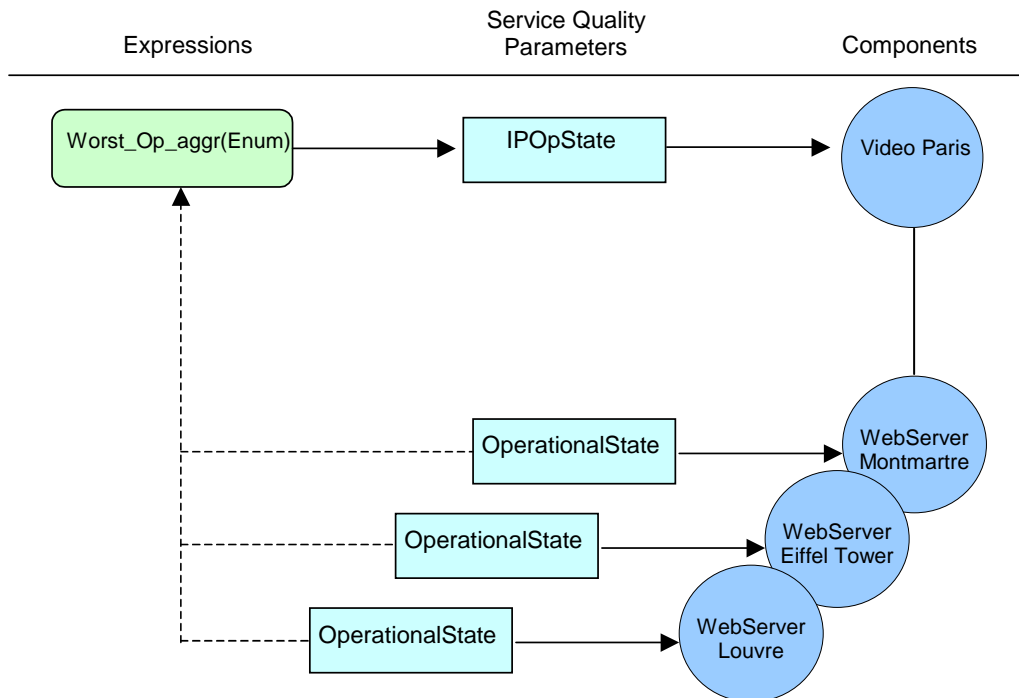
The service provider selects the Worst_Op_aggr(Enum) expression using the OpenView Service Designer UI as illustrated in Figure 10.

Figure 10   Example Aggregation Expression
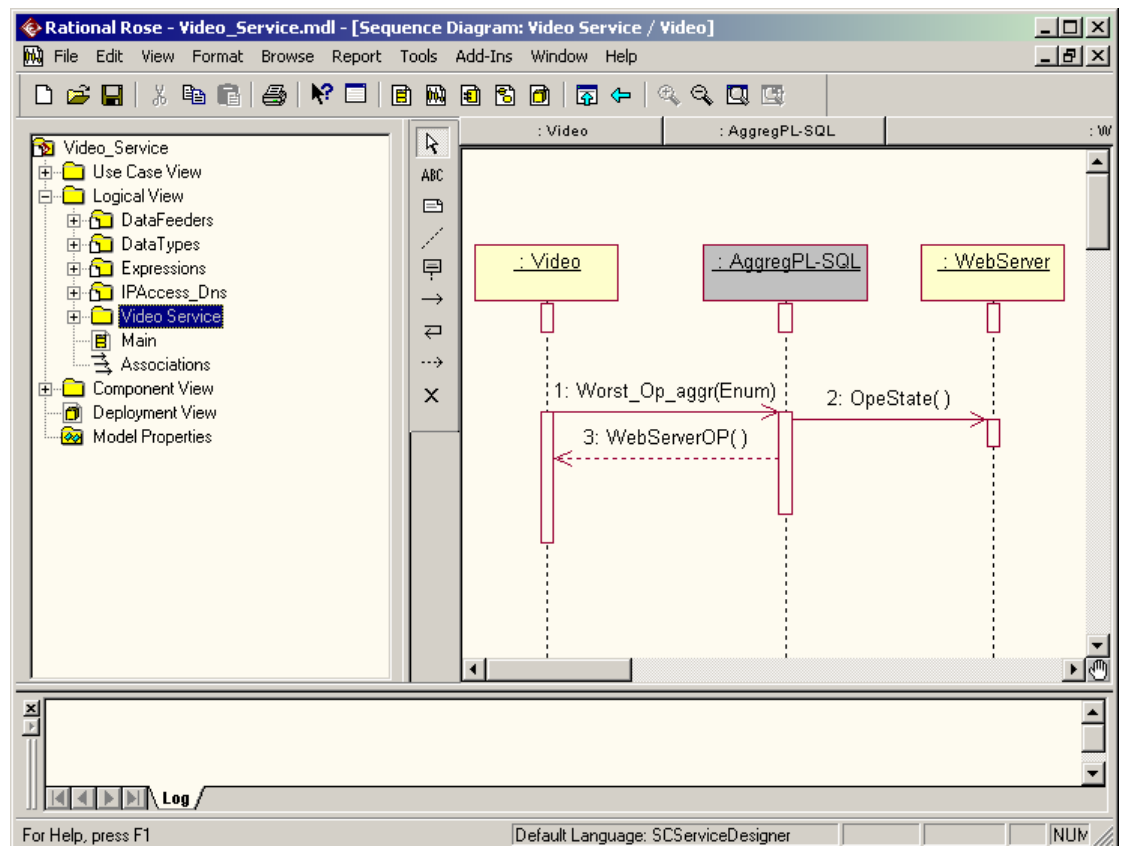
2.3 Implementing Your Service Model in XML

Once you have designed a service model, you need to implement it in XML. This section describes:

- Generating the XML code with the OpenView Service Designer UI.
- Loading the XML files in OpenView SQM.

2.3.1 Generating the XML Code with the OpenView Service Designer UI

The OpenView Service Designer UI generates the following files automatically:

- Data feeder XML files (one file for each data feeder).
- Expression calculation XML files (one file for each of your expression operations). Predefined expressions are already loaded in OpenView SQM. You can use them directly without generating them.
- Service XML files (one for each service).

The user does not need to change the data feeder XML files and the service XML files.

For an example of the XML file produced, refer to Acronyms.

2.3.2 Loading the XML Files

Before you can load the XML files, install the OpenView SQM platform with at least the core components. For information about installing OpenView SQM, refer to the *OpenView Service Quality Manager Installation Guide.*

The remainder of this section explains what you need to do before you load the files, the loading order, and the commands for loading them.

2.3.2.1 Before You Load the XML Files

To run the CLUI, you need to set the `CLUI_CFG_FILE` environment variable. This variable must be set to the configuration file that contains the TIBCO bus values that was automatically generated for you in the data directory.

2.3.2.2 Loading Order

You must load the XML files in the following order:

- Data feeder XML files.

- Expression XML files (only if you use custom expressions).

- Service model XML files

2.3.2.3 Commands for Loading XML Files

- To load a data feeder file called `PlatformDataFeeder.xml`, type the following command in the CLUI:

  `temip_sc_load_definition -d PlatformDataFeeder.xml`

- To enter an expression file called:`MyUserOperation_ExpressionCalculation.xml`, type the following command in the CLUI:

  `temip_sc_load_definition -e MyUserOperation_ExpressionCalculation.xml`

- To enter a service model file called `VideoService.xml`, type the following command in the CLUI:

  `temip_sc_load_definition -s VideoService.xml`

You can specify several files for each of the CLUI commands.

2.3.3 Example Service Model

An example of the model for the video service comes with the OpenView Service Designer UI installation kit. The example uses the structure described in Section 2.2.3 so that you can navigate through the diagrams and get familiar with the capabilities of the OpenView Service Designer UI.

2.3.4 Updating the Service Model

This section describes how to update the service model with new service components and new service parameters.

You can also update service definitions, data feeders, and expressions. Data feeders have a version, allowing you to use different versions of the same data feeder simultaneously. For information about the implications of updating service definitions, data feeders, and expressions, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide.*

2.3.4.1   Adding New Service Components

You can add new service components and incorporate them into an existing service model. For information about the implications of updating service definitions, data feeders, and expressions, refer to *OpenView Service Quality Manager Information Modeling Reference Guide*.

For example, the video service provider decides to host the video server on another machine dedicated to this task and monitor its activity separately. The service designer creates a video server component. This component needs to run on a platform, so the designer links it to the existing platform component.

Figure 11 illustrates the new model.

Figure 11   New Video Service Model
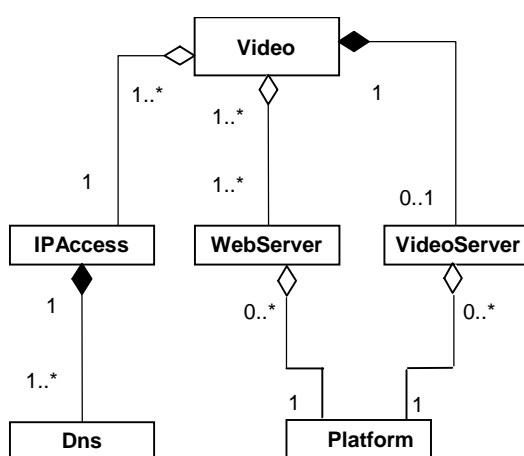
The `VideoServer` service component uses the MovieDuration and OperationalState parameters.

To create a new service component, the service provider needs to do the following:

- Add a new `VideoServer` service component and its parameters using the OpenView Service Designer UI as illustrated in Figure 12   Adding a VideoServer Component Using the Service Designer UI

- Create a new data feeder definition that contains the input parameters for the new service parameters using the OpenView Service Designer UI.

- Specify any expressions required to calculate the service parameters using the OpenView Service Designer UI.

- Update the SA configuration mapping to incorporate any new parameters. The SAs automatically reload the new data feeder definition. Stop and restart the OpenView SQM core components.

- Instantiate the new service component instance using the OpenView SLA Administration UI.

Figure 12   Adding a VideoServer Component Using the Service Designer UI
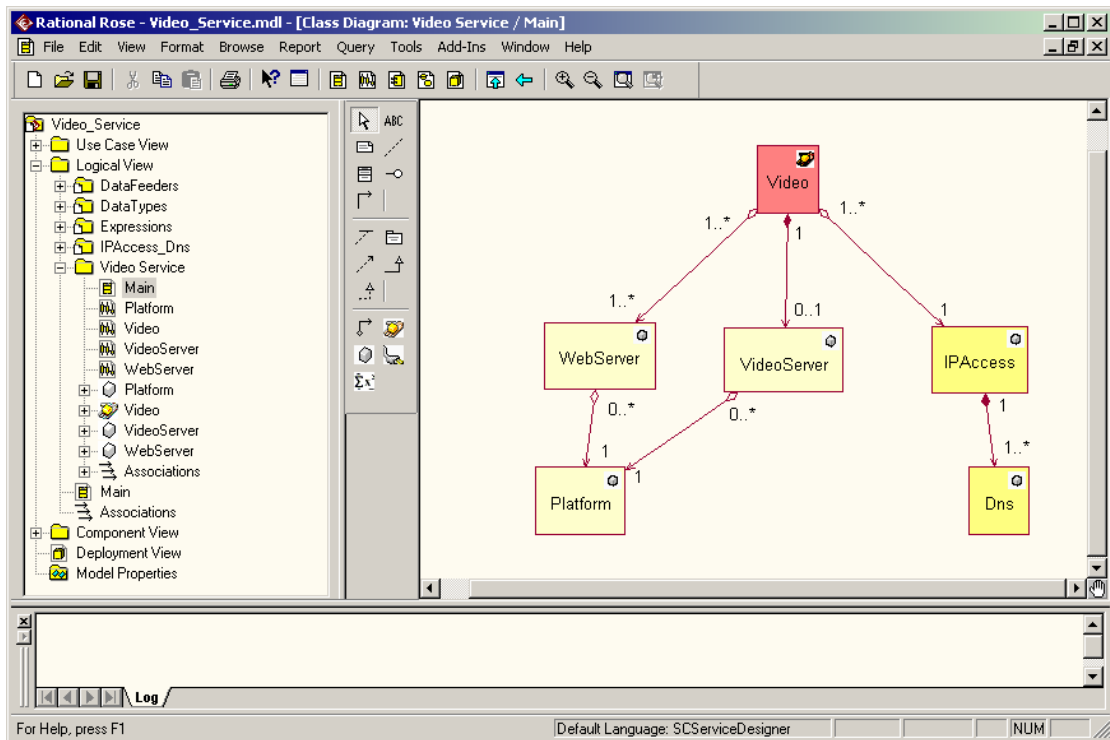
2.3.4.2   Adding New Service Parameters

You can add new service parameters and incorporate them into an existing service model. For information about the implications of updating service definitions, data feeders, and expressions, refer to *OpenView Service Quality Manager Information Modeling Reference Guide*.

For example, the video service provider decides to create a new `Bandwidth` service parameter that it will associate with the `platform` service component.

This example assumes the following:

- The data feeder used to derive the new service parameter already exists and must be modified to accommodate the new input parameter.

- The expression used to obtain the new service parameter is based on existing expressions.

The service provider uses the following procedure:

- Add a new service parameter, `bandwidth`, using the OpenView Service Designer UI.

- Add the corresponding data feeder parameter to a new version of an existing data feeder definition using the OpenView Service Designer UI.

- Add expressions using the OpenView Service Designer UI.

- Generate the updated XML files using the OpenView Service Designer UI.

- Ensure that the existing service adapter supports the new input parameter.

- Load the XML filed into OpenView SQM using the CLUI.

- Update the service adapter configuration mapping with the new service parameter mapping. The SA should automatically reload the new data feeder definition.

- Stop and restart the OpenView SQM core components.

Chapter 3

Instantiating a Service

This chapter describes the service instantiation phase of the SLA lifecycle. The service instantiation phase takes the model produced by the service design phase and creates specific service instances with OpenView SQM.

This chapter contains the following sections:

- Section 3.1 Before You Begin
- Section 3.2 Configuring Service Instances
- Section 3.3 Collecting Service Quality Data

3.1 Before You Begin

This section describes information you need before beginning the service instantiation phase of your service level monitoring solution.

3.1.1 Prerequisites of the Service Instantiation Phase

The service instantiation phase assumes that you have:

- Started OpenView SQM components and the OpenView SLA Administration UI.
- Designed the service.
- Installed the associated service adapters (optional).
- Ensured that the service infrastructure and OSS system involved are deployed and running (optional).

3.1.2 Goal of This Phase

After the service design phase, the service instance corresponds only to the service infrastructure or to the OSS involved with the service. In the service instantiation phase, a customer has subscribed to the new service and the service needs to have its quality managed. This phase typically occurs when:

- The OpenView SQM solution is deployed for the first time.
- A new service is deployed.

3.1.3 Overview of the Instantiation Process

The OpenView SLA Administration UI instantiates services. The service instantiation phase has two parts: configuration and data collection.

First, you must configure the service instance in OpenView SQM. You configure the service instance as follows:

- Install and start the service adapter. Refer to the corresponding service adapter installation guide for more information.

- Create service and service component instances.

- Register data feeders (only if the relevant service adapters are not installed).

- Bind data feeder instances with service component instances.

You can start collecting service quality data after configuring the service instance or creating the SLA. To collect service quality data, you must:

- Install and start the service adapter. Refer to the installation guide for the service adapter for more information.

- Check the collection status using the OpenView Administration UI.

The remainder of this chapter describes how to configure service instances and collect service data.

3.2 Configuring Service Instances

This section describes how to configure service instances, including registering data feeder instances, creating service component instances, and associating the data feeder instances with service component instances.

3.2.1 Creating Service Instances

Before you can instantiate service components and create SLAs, you must first instantiate the service itself. You create service instances with the OpenView SLA Administration UI, by selecting New/Service from the File menu. You can also create service instances directly using the Command-Line UI.

For example, the administrator of the video service creates an instance of the video service for customers in France as described in Figure 13.

Figure 13   Instantiating the Video-France Service
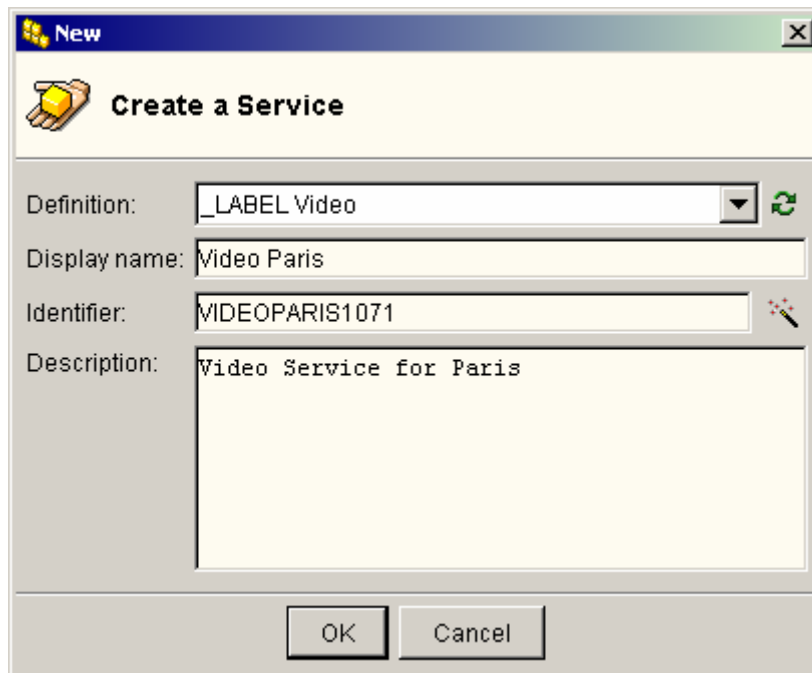
When the customer base of the ISP grows, they can create service instances that correspond to other geographical regions, such as `Video-London` for the United Kingdom and Video-`NewYork` for North America.

For information about updating the service object model after a service instance has been created, refer to Section 2.3.4.

3.2.2   Creating Service Component Instances

Next, you must instantiate each of the service components and provide the property information using the OpenView SLA Administration UI. You can also create service component instances directly using the Command-Line UI.

For example, the administrator of the video service creates an instance of the IPAccess and DNS service components. Figure 14 illustrates how the administrator instantiates the DNS service component (in this figure the IPAccess component is already instantiated).

For information about updating the service object model after a service component instance has been created, refer to Section 2.3.4.

Figure 14   Instantiating the IPAccess Service Component
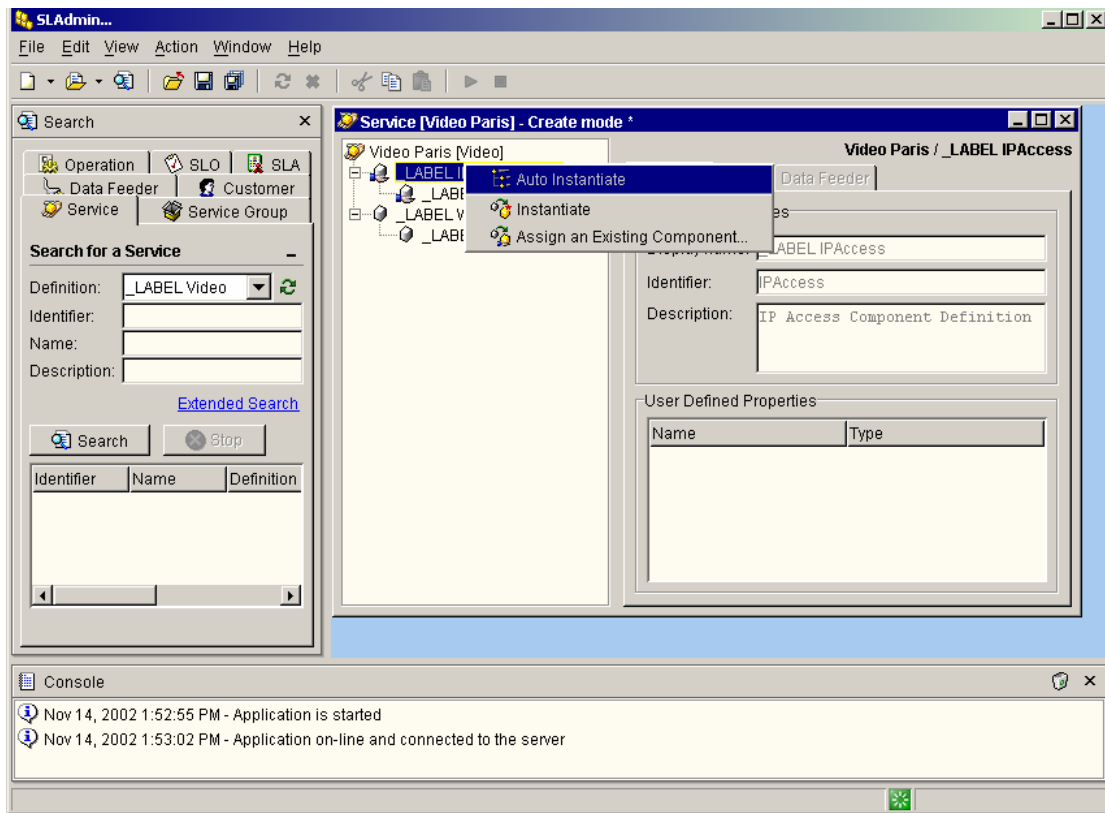
The administrator can also create component instance by duplicating an existing component instance. Figure 15 illustrates how the administrator duplicates the Web server instance of the service component by right clicking on the existing instance and selecting "Duplicate Component".

Figure 15   Duplicating a Service Component Instance

For more information about filling in the service component properties, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide*.

For information about updating the service object model after a service component instance has been created, refer to Section 2.3.4.

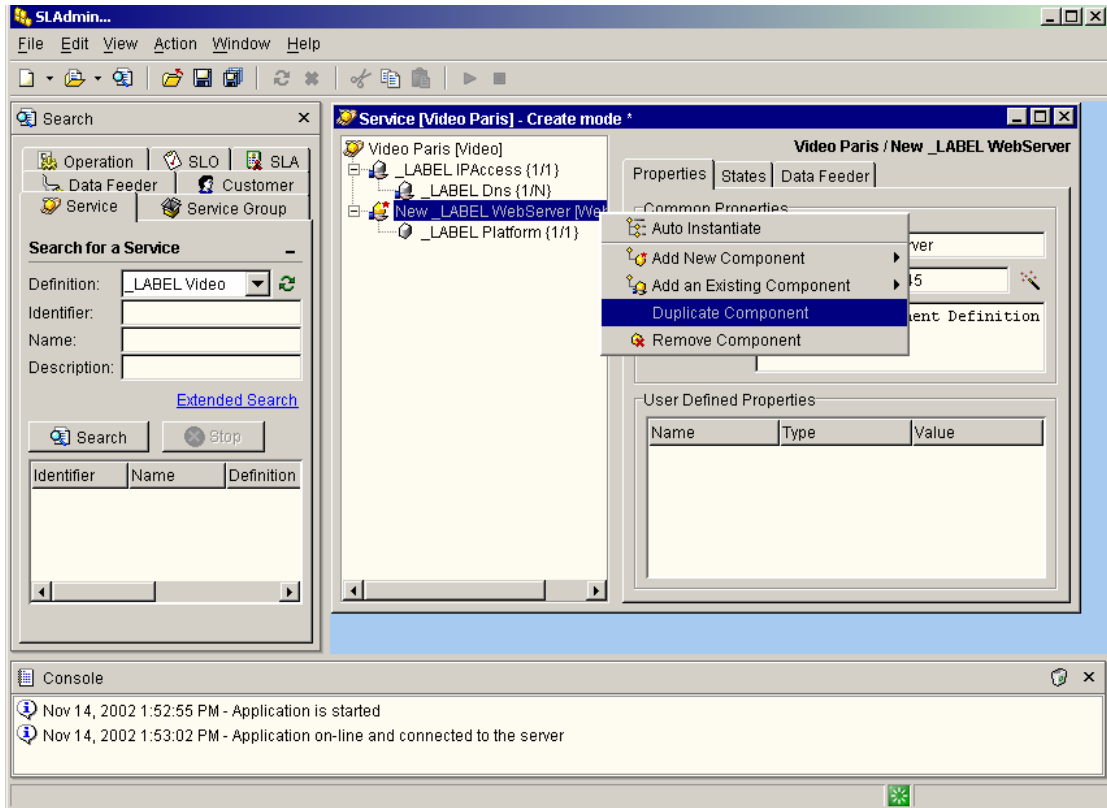## 3.2.3   Creating or Registering Data Feeder Instances

If you have not yet started the service adapters of the service instance, you must register the data feeders in OpenView SQM to proceed with the subsequent actions. You can also create data feeder instances directly using the Command-Line UI.

Make sure that you respect the MRP naming scheme. OpenView SQM needs to identify the correct data feeder using this MRP Naming Scheme.

For more information about registering data feeders, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide*.

For information about updating the service object model after a data feeder instance has been created, refer to Section 2.3.4.

3.2.4   Associating Service Component Instances with Data Feeder Instances

Each service component has to be associated with the appropriate data feeder, so that the correct service quality data from the data feeders arrives in the associated service component.

For example, the administrator of the video service associates the instance of the `Patrol_IpDF` data feeder registered in Section 3.2.1 with the `IPAccess` service component instance. Figure 16 illustrates this process.

Figure 16   **Associating a Service Component Instance to a Data Feeder**
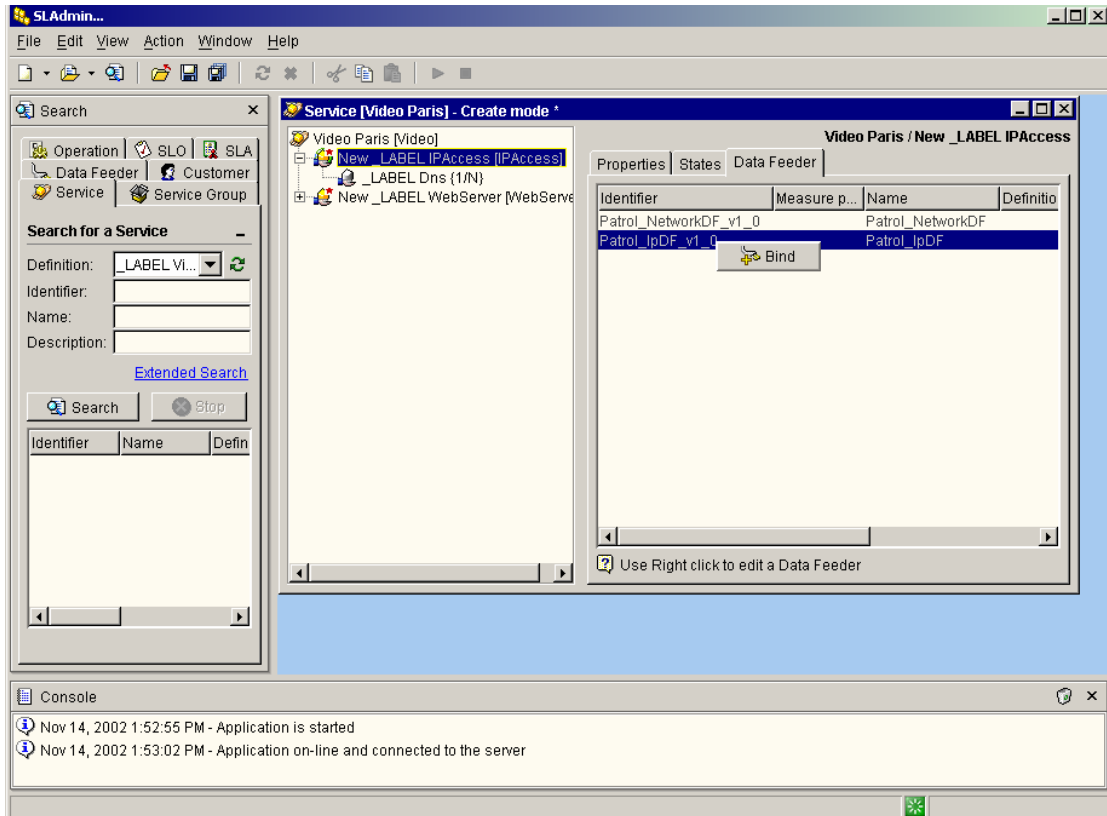
For more information about binding, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide*.

3.3 Collecting Service Quality Data

Service adapters collect service quality data from the parameters of data feeder instances. This section provides an overview of service adapters, describes how to install and launch them, and tells you how to check the collection status of the data feeders.

3.3.1 About Service Adapters

Service adapters (SAs) retrieve service quality data. An SA is a technology specific plug that can communicate with a service quality source.

OpenView SQM includes service adapters linked to HP products, including OV Service Navigator SA, OV Service Desk SA and the OV Internet Services SA. For other service adapters, contact your HP representative.

You must configure a service adapter to meet your specific requirements. The following information is included in the service adapter implementation or configuration. Make this information available to the operators that manage and install the SAs.

- The data feeder definitions supported by the SA. The data feeder definition describes the format of the service quality data in OpenView SQM.

- Data feeder instances targeted by the SA. Depending on the site of the measurement reference points, a service adapter can retrieve information about several data feeder instances. You must carefully evaluate an MRP naming scheme.

- Subscriber ID resolution. The subscriber ID resolution mechanism describes what element of the service quality data identifies the user of a service. You can use the subscriber ID to map the service quality data to the right customer. For more information about mapping subscriber IDs, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide.*

- Auto-discovery. Depending upon the product they connect with, some service adapters can automatically discover entities to manage and create corresponding data feeder instances.

3.3.2 Installing and Launching Service Adapters

For service quality data to be collected, you must install service adapters. For information about installing SAs, refer to the installation guide for your particular SA.

3.3.3 Checking the Collection Status

To confirm that data collection is proceeding as expected, you can check the collection status of the service instance. The operational state of the data feeder also provides information about data collection. When a data feeder is in the disabled state, the availability status provides information about why it is disabled.

For more information about checking the collection status, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide.*

Chapter 4

Creating Service Level Agreements

This chapter describes the SLA creation phase of the SLA lifecycle, where the service level agreements are associated with the service instances created in Chapter 3.

This chapter contains the following sections:

- Section 4.1 Before You Begin
- Section 4.2 Creating a Service Level
- Section 4.3 Creating Service Groups
- Section 4.4 Creating Customers
- Section 4.5 Mapping Customers to Subscriber IDs
- Section 4.6 Creating an SLA

4.1 Before You Begin

This section describes information you need before beginning the SLA creation phase of your service level monitoring solution.

4.1.1 Prerequisites of the SLA Creation Phase

Begin the SLA creation phase only after you have:

- Installed at least the core components of OpenView SQM and the OpenView SLA Administration UI.
- Designed the service.
- Created at least one instance of a service.
- Deployed the service infrastructure and OSS that support the service (optional but recommended).
- Installed the service adapters that the SLA needs (optional).

4.1.2 The Goal of This Phase

In the SLA creation phase, you link the service to a customer or an internal operational requirement. During this phase, you describe the service quality objectives using customer expectations or your own internal requirements.

4.1.3 Overview of the SLA Creation Process

The process to create an SLA relies on the Service Level Administrator UI. To create an SLA, perform the following tasks:

- Create service levels.

- Create service groups.

- Create customers that correspond to the customer or internal operation concerned by the SLA.

- Create the SLA, associating a particular service level with a customer.

The remainder of this chapter describes these tasks in detail.

Note

You update customers, service groups, service levels, and SLAs using the same procedures described in this chapter.

4.2 Creating a Service Level

A service level defines the thresholds needed to control the quality of a service sold to a customer. You create service levels objectives with the OpenView SLA Administration UI. Refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide* for details about creating service levels.

For example, the ISP wants to create a service level called Gold-Video that defines the threshold for the highest quality of the service they offer. They decide that the minimum bandwidth for customers of the Gold-Video service is 512 KB per second.

Component objectives are used to set the bandwidth thresholds. The service designer sets this objective using the OpenView SLA Administration UI. Figure 17 illustrates the component objective that sets the threshold for the minimum bandwidth.

Figure 17   Creating the Minimum Bandwidth Service Level
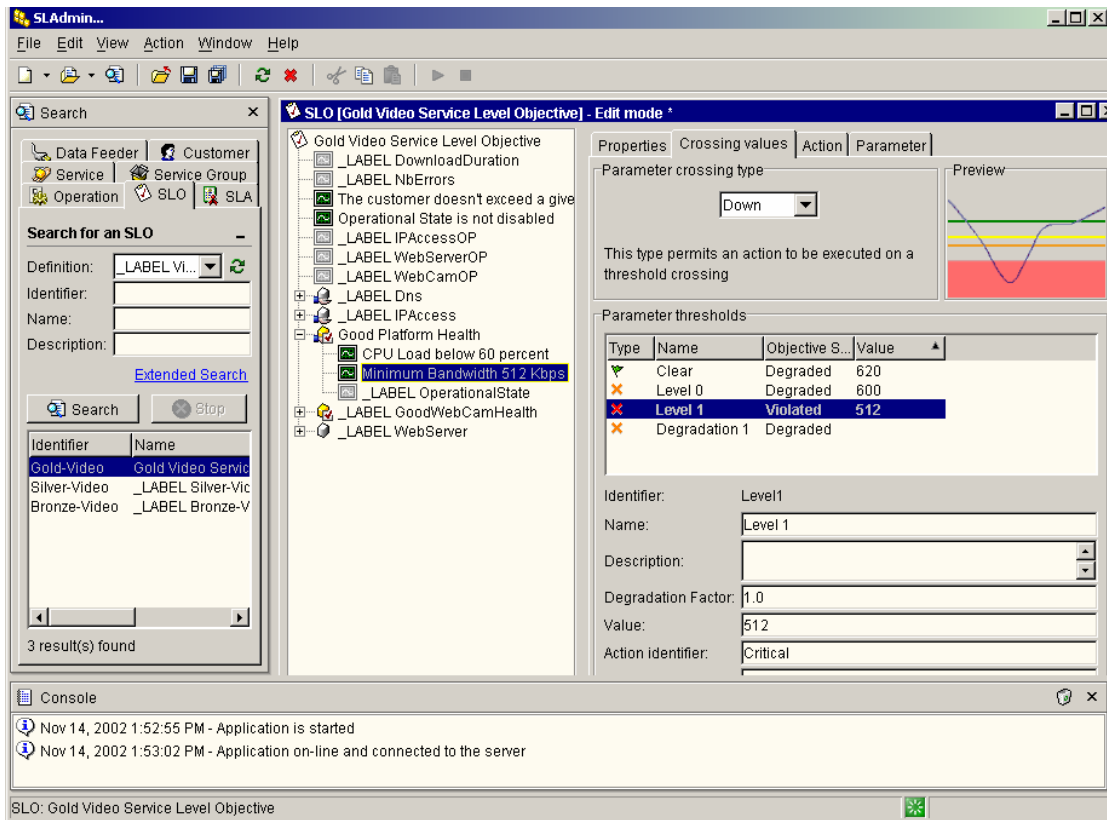
You can associate an OSI Alarm to a threshold-crossing event during an SLA violation.

4.3 Creating Service Groups

Service groups gather individual component instances into groups that you can later link with an SLA. To create service groups, you must use the OpenView SLA Administration UI.

For example, the video service provider decides to create a service group for customers in France. Once created, the administrator associates service instances with the service group by clicking the Services tab.

Figure 18 illustrates how the administrator selects the Paris video service to associate it with the France service group.

Figure 18   Associating a Service with a Service Group
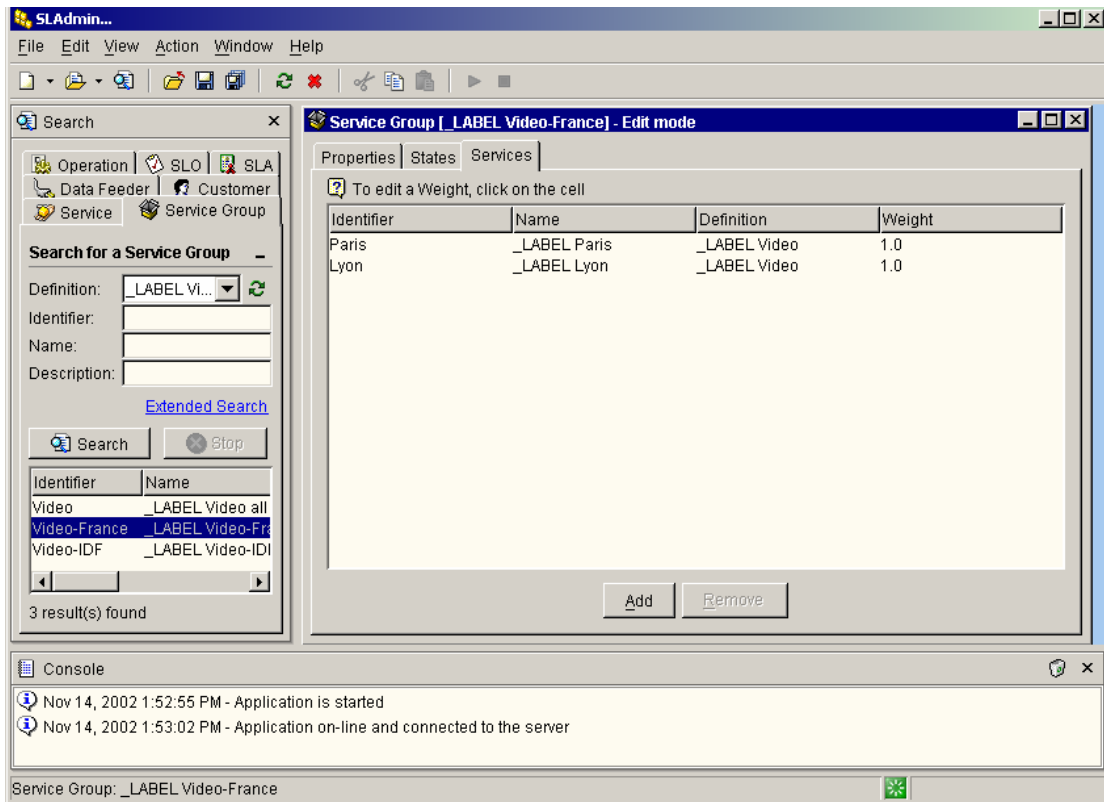
For more information about how to create service groups, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide*.

Later, when customers subscribe to the video service, you can add them to the service group.

4.4 Creating Customers

You create new customers with the OpenView SLA Administration UI. For example, the administrator for the video service creates a new customer, Software House A.

For more information about creating customers, refer to the *OpenView Service Quality Manager Service Level Administration Client User's Guide*.

4.5 Mapping Customers to Subscriber IDs

An ISP can offer services, such as mail and Web services, to other businesses. These businesses, in turn, can give employees and subscribers access to the services. For example, Software House A buys the Video service from the ISP.   Software House A then provides its employees access to this service. For the ISP to manage service quality, it needs to be able to map service interactions made by Software House A employees with the correct customer SLA.

To make this mapping between subscribers and customers, OpenView SQM uses subscriber IDs. The subscriber ID identifies the user of the service in a service

quality measure. The subscriber ID can be an IP address, a user identifier configured in a label, or an identifier directly linked with a user, such as a DSL Port number.

Sometimes, the user happens to be the customer. So, the subscriber ID maps directly to the customer ID. In other cases, the customer does not use the service. Subscribers have a contractual relationship with the customer so that they can use the service. So, several subscriber IDs map to one customer.

For some dedicated data feeders, the SA can automatically flag a premise subscriber ID. When flagged as a premise subscriber ID, the subscriber will always be the same. For example, a premise subscriber ID could correspond to equipment that forms part of the service infrastructure at the customer premises.

If no mapping is required, the domain is a generic "OpenView SQM" domain.

For information about writing a mapping file in XML and providing it to OpenView SQM, refer to the *OpenView Service Quality Manager Information Modeling Reference Guide.*

4.6   Creating an SLA

You create SLAs using the OpenView SLA Administration UI or directly using the command-line UI. The SLA associates service levels with customers and service groups.

For example, the video service administrator creates a customer SLA for Software House A, a customer created in section 4.4. Because Software House A has offices in France, it is part of the Video-France service group as described in section 4.2. The service level for this SLA is the Gold-Video service level created in section 4.2.

Once the SLA has been defined, the administrator associates it with the customer, service group, and service levels defined in the previous sections as illustrated in Figure 19.

Figure 19   Defining the Properties of the SLA
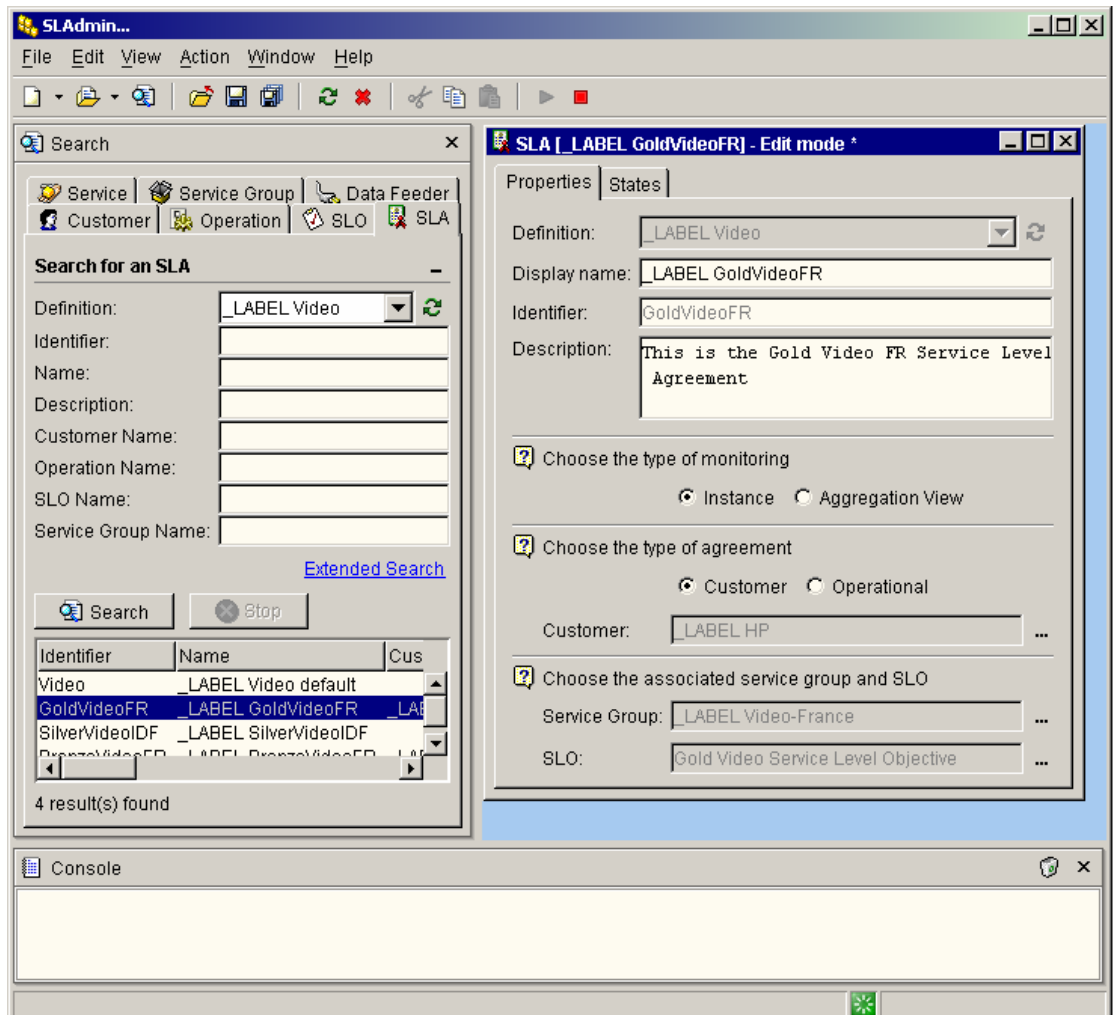
You can also create SLAs that measure service level compliance. Figure 20 illustrates how to create an SLA for measuring service level compliance.

Figure 20   Creating a Compliance SLA

Next, you use the OpenView SLA Administration UI to specify a compliance period, time, and in-service.

Figure 21   Specifying Service Level Compliance Hours
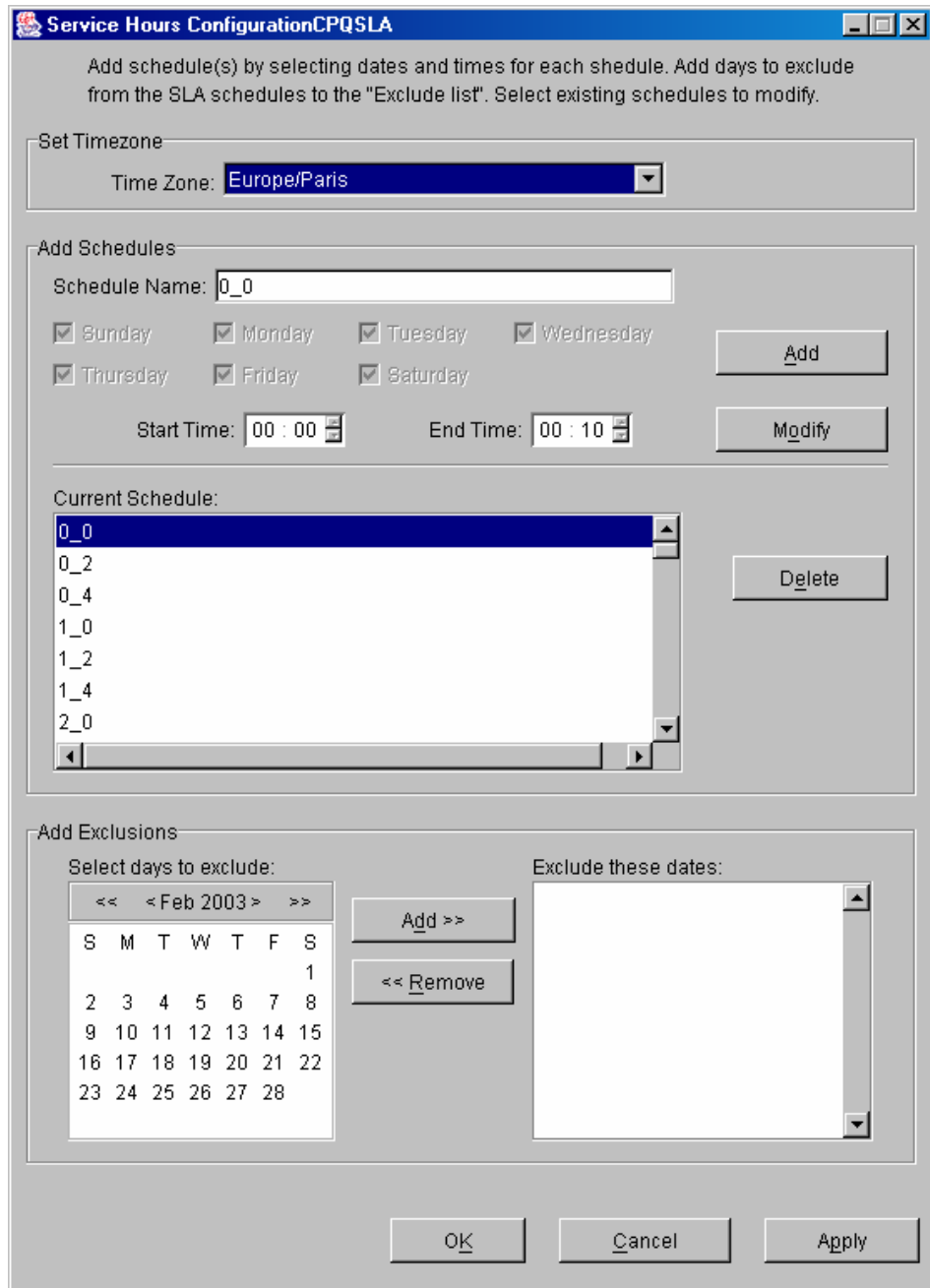
For more information about service level compliance, refer to the *OpenView SLA Administration User's Guide.*

Chapter 5

Monitoring the Service Level

Once you have carried out the first three phases of the SLA lifecycle described in the previous chapters, you can begin to use OpenView SQM to monitor service quality.

This chapter contains the following sections:

- Section 5.1 Before You Begin

- Section 5.2 Monitoring Service Quality

5.1 Before You Begin

This section describes information you need before beginning the service monitoring phase of your service level monitoring solution.

5.1.1 Prerequisites of the Service Monitoring Phase

Begin the service monitoring phase only after you have completed the following:

- Installed OpenView SQM.

- Designed your service.

- Instantiated at least one service instance.

- Specified at least one SLA.

- Collected service quality data.

5.1.2 The Goal of This Phase

In the service monitoring phase, your service is running and has been provided to users according to criteria specified in an SLA. This phase allows you to monitor any variation in service quality so that you can react quickly to any problems.

5.2 Monitoring Service Quality

Use the OpenView SLA Monitoring UI to monitor service quality. To monitor service quality you need to:

- Select one or more SLAs to monitor from a set of criteria.

- Visualize the state of the monitored SLAs.

- Visualize the state of each service instance, service component instance, and parameter.

- Customize charts that allow you to visualize in real-time the value of parameters you select.

- Customize a dashboard that can be saved and retrieved the next time you connect to the SLA Monitoring UI.

- Launch the reporting application to view the historical data of the parameter you have selected.

For details about each of the above tasks, refer to the *HP OpenView Service Quality Manager SLA Monitoring User Interface User's Guide*.

5.2.1   About the OpenView SLA Monitoring UI

The OpenView SLA Monitoring UI connects to the SQM components through the Presentation Server. Once connected to this server, you select the SLA or SLAs that you want to monitor using the Scope Definition dialog box. Figure 22 illustrates how an administrator defines the SLA scope for the video service created in the previous chapters.

Figure 22   Defining the SLA Scope
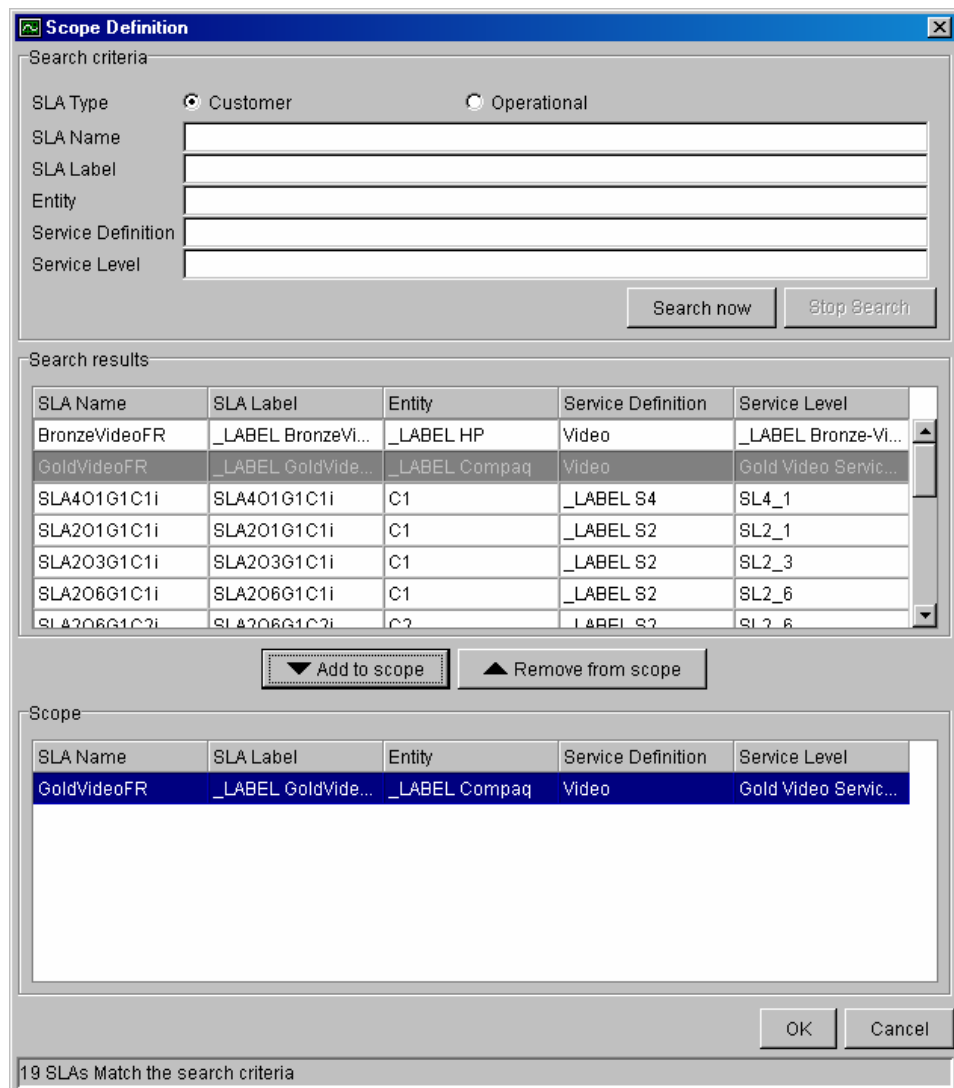
The administrator selects the SLA to be monitored, `GOLDVideoFR`, and the SLA monitoring main window opens. It contains the following information:

- The SLA to be monitored, along with its state, type, name, and entity.

- A tree view of the SLA, which displays the status of compliance objectives (from the SLA to the service component).

- From these windows you can access different "Details" windows that provide static information about the SLA and its service components.

Figure 23 gives an example of the information that the administrator sees after selecting the GOLDVideo_FR SLA.

Figure 23   SLA Monitoring UI
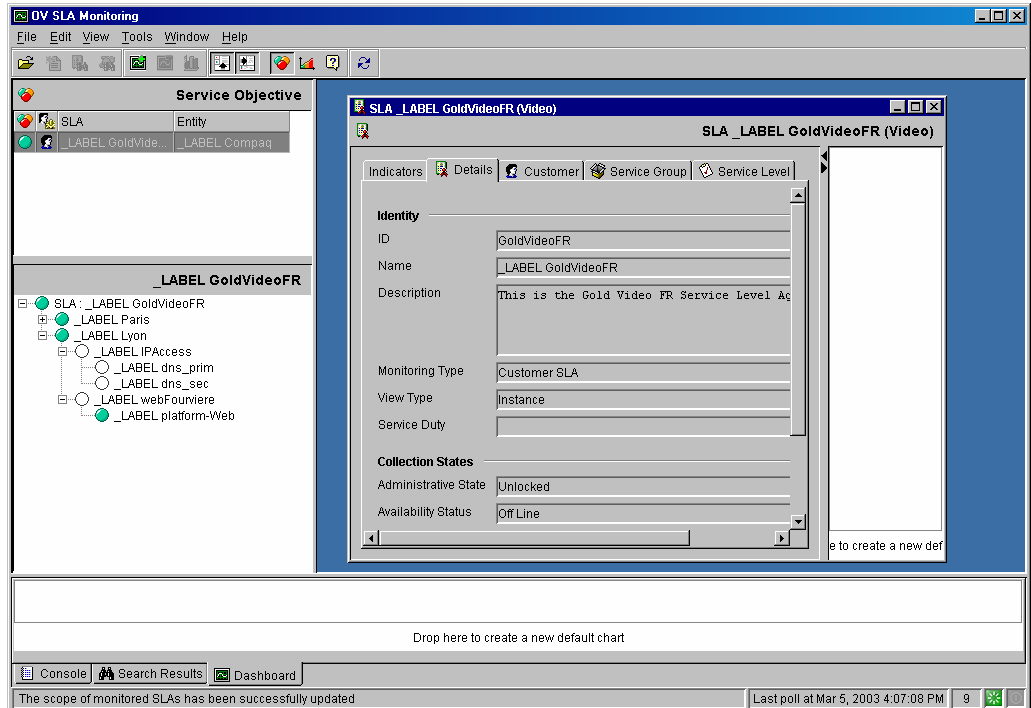
You can also use the SLA Monitoring UI to track the affect of a degradation or a violation and to view the state of SLAs, service instances, service components, and parameters.

For example, to monitor the OperationalState of the video service, the administrator selects the Video-Paris instance and adds a chart in the Dashboard. The administrator can then see the evolution of this parameter over time.

Each time the parameter crosses a threshold, the color of the corresponding status icon will change.

Figure 24 shows the evolution of the OperationalState during the last two hours.

Figure 24   Monitoring the OperationalState Parameter
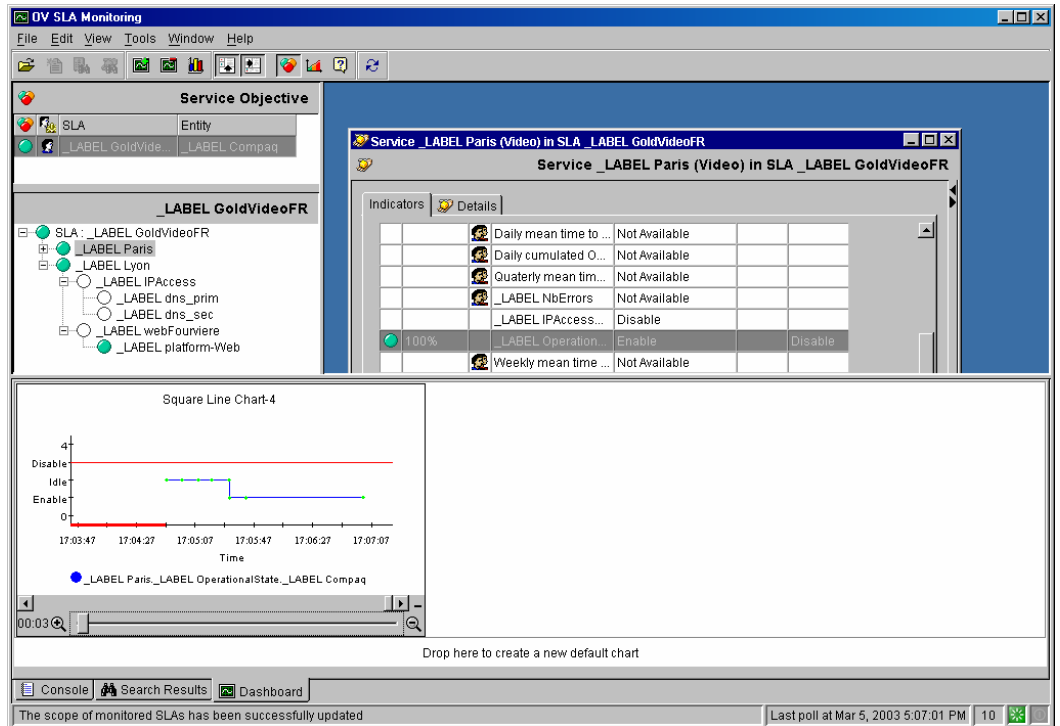
For information about the other tasks you can perform with the SLA Monitoring UI, see the *OpenView Service Quality Manager SLA Monitoring User Interface User's Guide*.

Chapter 6

Creating Service Level Reports

Using the data collected during the service monitoring phase, you can now create service quality reports.

This chapter contains the following sections:

- Section 6.1 Before You Begin
- Section 6.2 Consulting SQM Reporting Reports
- Section 6.3 Using the Predefined Reports
- Section 6.4 Creating Custom Reports

6.1 Before You Begin

This section describes information you need before beginning the service reporting phase of your SQM solution.

6.1.1 Prerequisites of the Service Reporting Phase

Complete the following tasks before beginning the service reporting phase:

- Install OpenView SQM.
- Design the service.
- Create at least one service instance.
- Specify at least one SLA.
- Collect service quality data.

6.1.2 The Goal of This Phase

In the service reporting phase, you create reports that highlight service quality trends and overall SLA performance.

6.1.3 Who Should Use SQM Reporting?

The following SQM users may be interested in creating and using reports:

- **Operators**. An operator may want to see customer inventory reports, customer degradation and violation events reports, health reports (for SLAs, customer SLAs, and service), and performance reports.
- **Account managers**. Customer account managers may need customer inventory notice board reports, customer degradation and violation events reports, and health reports for customer SLAs and services.

- **Customers**. Customers can access the data concerning themselves, including inventory reports, degradation and violation events reports, and their own SLA and service availability reports.

- **Service operation center (SOC) operator**. SOC operators need access to service-relative reports, such as service provider inventory reports and performance reports.

- **SOC operations managers**. SOC operations managers need access to degradation and violation event reports, SLA service availability reports, and performance reports.

- **Users of custom SQM reports**. Users that create custom SQM reports need access to all the data while creating a new universe or a new report.

6.2 Consulting SQM Reporting Reports

You can consult the reports provided by SQM Reporting as follows:

- Using menus in the SLA Monitoring UI. The information available depends upon the context. For more information about monitoring services, see Chapter 5, Monitoring the Service Level.

- Using a Web browser that is connected to the WEBINTELLIGENCE server. BUSINESSOBJECTS InfoView is used to access all of the possible reports.

- Using the BUSINESSOBJECTS Explorer. The Explorer provides access to BUSINESSOBJECTS and WEBINTELLIGENCE reports. However, all WEBINTELLIGENCE reports appear in the Explorer using the standard BUSINESSOBJECTS document template.

6.3 Using the Predefined Reports

You can use OpenView SQM Reporting to:

- Consult service health and performance data through BUSINESSOBJECTS and WEBINTELLIGENCE reports.

- Create new BUSINESSOBJECTS and WEBINTELLIGENCE reports.

A predefined set of BUSINESSOBJECTS and WEBINTELLIGENCE reports is provided with OpenView SQM Reporting, including reports about inventory statistics, service violations, service degradations, service health indicators, and performance data.

For example, Figure 25 illustrates an extract of "SLA Service Health" report that gives the Service Health History upon a given period of time.

Figure 25   Predefined Report Example

Service Health History   Service Events:

	March 22		March 23	
SA%	33.5646%	N/A	16.7326%	- 16.83
Service Downtime	0 days 04:07:25	N/A	0 days 06:39:40	+ 0 days 02:32:15
Number Of Unplanned Outages	0		0	
Service QoS Degradtion	0 days 02:05:00	N/A	0 days 01:20:19	- 0 days 00:44:40
Number Of QoS Degradations	74		29	
Service Uptime	0 days 00:00:00	N/A	0 days 00:00:00	0 days 00:00:00
MTBF	0 days 00:00:00	N/A	0 days 00:00:00	+ 0 days 00:00:00
MTTR	0 days 00:03:20	N/A	0 days 00:13:46	+ 0 days 00:10:26

Cumulative Failure/Degradation Duration

Duration (min): 400, 350, 300, 250, 200, 150, 100, 50, 0

3/22/05   3/23/05

SLA QoS degradation duration   ■ SLA outage duration

6.4 Creating Custom Reports

You can create a new report using the generic universe (Master.unv) or using a specialized universe that you create. You can create custom reports as follows:

- By creating a new universe using the BUSINESSOBJECTS designer.

- By creating new BUSINESSOBJECTS and WEBINTELLIGENCE reports.

The following sections describe each customization in detail.

6.4.1 Creating a Specialized Universe

Specialized BUSINESSOBJECTS universes are created by modifying a generic universe with service definition, service component definition, and service parameter specific objects.

Note

You need basic Structure Query Language (SQL) and ORACLE knowledge to create new universes. You also need to be familiar with BUSINESSOBJECTS Designer concepts, including universe, loop resolution with alias tables and contexts, the AggregateAware feature, and the hierarchies' behavior.

To create a specialized universe, you need to:

- Create a new universe, which is a copy of a static universe.

- Remove objects that you do not use and only keep the classes that apply to your service.

- Create a new class that contains your service and service component definition specific dimensions as well as detail and condition objects. Insert the new ORACLE tables to contain the definition and object information.

- Create a new class to contain your service and service component definition measures and insert the new ORACLE tables.

- Publish your universe with a new name.

6.4.2 Creating New Reports

You can create new BUSINESSOBJECTS and WEBINTELLIGENCE reports. These new reports can be created using the templates delivered with the reports. To make the report available to other users, you have to publish it.

A custom new BUSINESSOBJECTS report can be based on several universes. A custom WEBINTELLIGENCE report contains only one chart or one table. These reports are based on a single universe.

For your custom reports to be accessible from the SQM SLA Monitoring UI, the report must use predefined prompts and be published in a specific category.

You can define the following information in your new report:

- Report prompts

- Filters

- Drillable report levels

- Report categories

- Default reports for a specific category

For more information about creating reports, see the *OpenView SQM Reporting and Customization User's Guide* and the *OpenView SQM Datamart User's Guide*.

Acronyms

The following table describes the acronyms commonly used in this document:

Term	Description
ATM	Asynchronous Transfer Mode
BI	Business Intelligence
BO	Business Object
BSS	Billing Support System
CNM	Customer Network Management
CRM	Customer Relationship Management
DC	Data Collector
DFD	Data Feeder Definition
DFI	Data Feeder Instance
DMZ	Demilitarized Zone
EAI	External Application Interface
GUI	Graphical User Interface
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
MRP	Measurement Reference Point
MSL	Management Specification Language
MTBF	Mean Time Between Faults
MTTR	Mean Time To Repair
MVNO	Mobile Virtual Network Operator
NOC	Network Operation Center
OS	Operating System
OSI	Open Systems Interconnection
OSA	Open Service Architecture
OSS	Operation Support System
QoS	Quality of Service
SA	Service Adapter

Term	Description
SAI	Service Adapter Instance
SAP	Service Access Point
SC	Service Component
SD	Service Definition
SI	Service Instance
SIA	Service Impact Analysis
SIM	Service Integration Map
SLA	Service Level Agreement
SLM	Service Level Management
SLO	Service Level
SNMP	Simple Network Management Protocol
SOC	Service Operation Center
SPD	Service Parameter Definition
SPDM	Service Performance Data Manager
SR	Service Resource
SRM	Service Repository Manager
SQL	Standard Query Language
TOM	Telecommunications Operation Map
UI	User Interface
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
XML	eXtensible Mark-up Language
WAP	Wireless Access Protocol

Glossary

This glossary defines terminology commonly used in HP OpenView Service Quality Manager.

Auto instantiate (SLA Administration)

This action automatically creates an Instance of the Object selected. When the instance is created, the initial values of its instance variables are assigned.

BI

See business intelligence.

Business intelligence (BI)

A broad category of applications and technologies for gathering, storing, analyzing, and providing access to data that helps users make better business decisions.

Collected binding

Describes how collected parameters are filled from measurement parameters: either directly assigned or through a more complex expression.

Collected parameters

Known as KPI in the TMF, they represent the parameters collected from the Service Adapters (*measurement parameters*) and mapped into SQM *service component* parameters.

Computed binding

Describes how computed parameters are filled from collected parameters: either directly assigned or through a more complex expression.

Computed parameters

Known as KQI in the TMF, they represent the parameters calculated from *collected parameters*.

CNM

See customer network management

Customer

Companies or organizations that make use of the *services* offered by a *service provider*, based on a contractual relationship.

Customer network management

Customer network management is enabled by means of tools that provide business customers with access to management information originating from the service provider.

Data collection interval

The interval of time over which performance parameters are retrieved from the monitored service resources. This interval does **not** have to be the same as the *measurement interval* because *service adapters* or service resources may buffer statistics.

Data feeder

OpenView Service Quality Manager's source of data. A data feeder models service resources by defining one or more service parameters.

Data feeder definition

The static definition of a data feeder that models service resources by defining one or more service parameters.

Degraded service

The presence of anomalies or defects that cause degradation of the *quality of service*, but do not result in the total failure of the *service*.

Instantiate (SLA Administration)

Instantiate differs from Auto Instantiate in that items are instantiated individually.

Measurement interval

The interval of time over which each service parameter is measured. For example, a parameter may be the number of discarded packets, measured over a 15-minute measurement interval.

Measurement parameters

They represent the parameters directly collected by the Service Adapters. These parameters are defined in the *Data Feeders*.

Measurement Reference Point (MRP) naming scheme

This is the formal description of how the measurement point name is built, that is, by concatenating the values of Data Feeder properties and fixed strings.

Mobile virtual network operator

A mobile operator that does not own its own spectrum and usually does not have its own network infrastructure. Instead, MVNOs have business arrangements with traditional mobile operators to buy **minutes of use** for sale to their own customers.

MRP

See Measurement Reference Point.

MVNO

See mobile virtual network operator.

Parameter

A value or set of values that are periodically updated and that help determine the quality of service.

Parameter objective

A set of objectives for the parameters belonging to a service.

Property

Special static parameters that are given a value only when an instance of an OpenView Service Quality Manager **Object** is created. For example, a Service Component can have a property called "location".

QoS

See quality of service.

Quality of service (QoS)

The ITU-T has defined quality of service as "the collective effect of service performances that determine the degree of satisfaction of a user of the service".

Service

A Service is a set of independent functions (Service Components) that consist of hardware and software elements and an underlying communications medium. A Service can include anything from a single leased-line service, to a complex application, such as vision conferencing.

Service availability

A measurement made in the context of a *service level agreement* that is expressed as a percentage. This percentage indicates the time during which the *service* is operational at the respective *service access points*.

ServiceCenter Repository

The ServiceCenter Repository is the storage center for all Service Quality Manager data. It receives data from the various Service Quality Manager interfaces and each interface can request information from the Repository.

Service component

An independent function that is part of a *service*, such as a hardware or software element, or the underlying communications medium.

Service component instance

The instance of a Service Component Definition that is active in the network, such as an instance of the IPAccess Service Component definition called "pop".

Service level (SL)

Defines Service Parameters and operational data enforced by the Service Level Agreement (for example, Max Jitter < 10 ms).

Service level agreement (SLA)

There are two type of Service Level Agreement, the **Customer** Agreement: a contract between a *service provider* and a *customer*, which specifies in measurable terms what the service provider supplies to its customers, and the Operational Service Level Agreement, which specifies in measurable terms the operational levels of the Service. A *service level agreement* is composed of individual objectives.

Service level objective (SLO)

The set of objectives for the parameters belonging to a Service or Service Component.

Service parameter

See *parameter*.

Service provider

A company or organization that provides *services* as a business. Service providers may operate networks or may integrate the services of other providers.

Service instance (SI)

The instantiated service definition that is active in the network, such as an instance of the video service definition called "Paris".

Service instance group (SIG)

A **group** of *service instances* against which the *service availability* must be reported. Each *service instance* belongs to one or more Service Instance Groups and each SIG contains at least one Service Instance. The relationship between the SIG and the Service Instances is defined in their *service level agreement*.

Service quality parameters

Service quality parameters represent computed and collected parameters.

SI

See Service Instance.

SIG

See Service Instance Group.

SL

See Service Level.

SLA

See Service Level Agreement.

SLO

See service level objective.

Subscriber

The entity responsible for the payment of charges incurred by one or more users.

User

An entity designated by a customer to use the services of a telecommunication network, such as a person using a UMTS mobile station as a portable telephone.